VLSI Design of Neural Networks

edited by

Ulrich Ramacher

Ulrich Rückert

Kluwer Academic Publishers

VLSI DESIGN OF NEURAL NETWORKS

edited

by

Ulrich Ramacher

Corporate Research & Development, SIEMENS AG

and

Ulrich Rückert

University of Dortmund

KLUWER ACADEMIC PUBLISHERS

Boston/Dordrecht/London

Distributors for North America:
Kluwer Academic Publishers
101 Philip Drive
Assinippi Park
Norwell, Massachusetts 02061 USA

Distributors for all other countries:
Kluwer Academic Publishers Group
Distribution Centre
Post Office Box 322
3300 AH Dordrecht, THE NETHERLANDS

Library of Congress Cataloging-in-Publication Data

VLSI design of neural networks / edited by Ulrich Ramacher and Ulrich Rückert.
p. cm. — (The Kluwer international series in engineering and computer science. VLSI, computer architecture, and digital signal processing)
Includes bibliographical references and index.
ISBN 0-7923-9127-6
1. Integrated circuits—Very large scale integration—Design and construction. 2. Neural networks (Computer science)—Design and construction. I. Ramacher, Ulrich. II. Rückert, Ulrich. III. Series.
TK7874.V559 1991
621.39'5-dc20 90-48757
CIP

Printed on acid-free paper.

Printed in the United States of America

VLSI DESIGN OF NEURAL NETWORKS

THE KLUWER INTERNATIONAL SERIES IN ENGINEERING AND COMPUTER SCIENCE

VLSI, COMPUTER ARCHITECTURE AND DIGITAL SIGNAL PROCESSING

Consulting Editor
Jonathan Allen

Wafer Level Integrated Systems: Implementation Issues, S.K. Tewksbury, ISBN: 0-7923-9003
The Annealing Algorithm, R.H.J.M. Otten, L.P.P. van Ginneken, ISBN: 0-7923-9022-9
Unified Methods for VLSI Simulation and Test Generation, K.T. Cheng, V.D. Agrawal, ISBN: 0-7923-9025-3
ASIC System Design With VHDL: A Paradigm, S.S. Leung, M.A. Shanblatt, ISBN: 0-7923-9032-6
BiCMOS Technology and Applications, A. R. Alvarez, Editor ISBN: 0-7923-9033-4
Analog VLSI Implementation of Neural Systems, C. Mead, M.Ismail (Editors), ISBN: 0-7923-9040-7
The MIPS-X RISC Microprocessor, P. Chow. ISBN: 0-7923-9045-8
Nonlinear Digital Filters: Principles and Applications, I . Pitas, A.N. Venetsanopoulos, ISBN: 0-7923-9049-0
Algorithmic and Register-Transfer Level Synthesis: The System Architect's Workbench, D.E. Thomas, E.D. Lagnese, R.A. Walker, J.A. Nestor, J.V. Ragan, R.L.Blackburn, ISBN: 0-7923-9053-9
VLSI Design for Manufacturing: Yield Enhancement, S.W..Director, W. Maly, A.J. Strojwas, ISBN: 0-7923-9053-7
Testing and Reliable Design of CMOS Circuits, N.K. Jha, S. Kundu, ISBN: 0-7923-9056-3
Hierarchical Modeling for VLSI Circuit Testing, D. Bhattacharya, J.P. Hayes, ISBN: 0-7923-9058-X
Steady-State Methods for Simulating Analog and Microwave Circuits, K. Kundert, A. Sangiovanni-Vincentelli, J. White, ISBN: 0-7923-9069-5
Introduction to Analog VLSI Design Automation, M. Ismail, J. Franca, ISBN: 0-7923-9102-0
Gallium Arsentide Digital Circuits, O. Wing, ISBN: 0-7923-9081-4
Principles of VLSI System Planning, A.M. Dewey ISBN: 0-7923-9102
Mixed-Mode Simulation, R. Saleh, A.R. Newton, ISBN: 0-7923-9107-1
Automatic Programming Applied to VLSI CAD Software: A Case Study, D. Setliff, R.A. Rutenbar, ISBN: 0-7923-9112-8
Models for Large Integrated Circuits, P. Dewilde, Z.Q. Ning ISBN: 0-7923-9115-2
Hardware Design and Simulation in VAL/VHDL, L.M. Augustin, D.C..Luckham, B.A.Gennart, Y.Huh, A.G.Stanculescu ISBN: 0-7923-9087-3
Subband Image Coding, J. Woods, editor, ISBN: 0-7923-9093-8
Low-Noise Wide-Band Amplifiers in Bipolar and CMOTechnologies,Z.Y.Chang, W.M.C.Sansen, ISBN: 0-7923-9096-2
Iterative Identification and Restoration Images, R. L.Lagendijk, J. Biemond ISBN: 0-7923-9097-0

CONTRIBUTING AUTHORS

P. Jespers
M. Verleysen
Univ. Cathol. de Louvain,Belgium

Ch. Nielsen
J. Staunstrup
Univ. of Lyngby,Denmark

J. Ouali
G. Saucier
INPG/CSI,Grenoble,France

P. Bessiere
LGI/IMAG,Grenoble,France

A. Chams
A. Guerin
J. Herault
C.Jutten
J.C. Lawson
LTIRF/INPG,Grenoble,France

J. Trilhe
SGS-THOMSON,Grenoble,France

P.Y.Alla
J.D.Gascuel
J.Roman
M.Weinfeld
Ecole Polytechnique,
Palaiseau,France

G.Dreyfus
A.Johannet
L.Personnaz
Ecole Supérieure,Paris,France

S. J. Prange
H. Klar
Techn. Univ. of Berlin,FRG

U. Rückert
Univ. of Dortmund,FRG

M. Kespert
P. Richert
M. Schwarz
FhG-IMS Duisburg,FRG

J. Nijhuis
A. Siggelkow
L. Spaanenburg
IMS Stuttgart, FRG

J. Anlauf
J. Beichter
N. Brüls
U. Hachmann
W. Raab
U. Ramacher
B. Schürmann
M. Wesseling
SIEMENS,Corp.R&D,FRG

J. Hoekstra
Delft University,The Netherlands

M. Chevroulet
E. Dijkstra
M.A. Maher
O. Nys
H. Oguey
E. Vittoz
CSEM,Neuchâtel,Switzerland

D. J. Baxter
S. Churcher
A. Hamilton
A. F. Murray
H. M. Reekie
Univ. of Edinburgh,U.K.

M. Brownlow
L. Tarassenko
Univ. of Oxford,U.K.

S. Jones
K. Sammut
Univ. of Nottingham,U.K.

J. A. Vlontzos
Siemens Corp. Research,
Princeton,USA

S. Y. Kung
Princeton Univ.,USA

FOREWORD

The early era of neural network hardware design (starting at 1985) was mainly technology driven. Designers used almost exclusively analog signal processing concepts for the recall mode. Learning was deemed not to cause a problem because the number of implementable synapses was still so low that the determination of weights and thresholds could be left to conventional computers.

Instead, designers tried to directly map neural parallelity into hardware. The architectural concepts were accordingly simple and produced the so-called interconnection problem which, in turn, made many engineers believe it could be solved by optical implementation in adequate fashion only. Furthermore, the inherent fault-tolerance and limited computation accuracy of neural networks were claimed to justify that little effort is to be spend on careful design, but most effort be put on technology issues.

As a result, it was almost impossible to predict whether an electronic neural network would function in the way it was simulated to do. This limited the use of the first neuro-chips for further experimentation, not to mention that real-world applications called for much more synapses than could be implemented on a single chip at that time.

Meanwhile matters have matured. It is recognized that isolated definition of the effort of analog multiplication, for instance, would be just as inappropriate on the part of the chip designer as determination of the weights by simulation, without allowing for the computing accuracy that can be achieved, on the part of the user.

Hardware design of neural networks is now more frequently preceded by the investigation of the constraints introduced by the application and system environment, the performance required and the comparison of various technologies. Designers strive for architectural solutions of the interconnection problem, trying to establish to what extent it is possible to deviate from the massively parallel networking and yet to satisfy an application-specific real-time constraint.

The papers presented in this volume reflect this new sight and mark the entry to a more sophisticated era of VLSI design of neural networks.

Ulrich Ramacher

Ulrich Rückert

PREFACE

The book originated from a workshop on MICROELECTRONICS FOR NEURAL NETWORKS which was held at the University of Dortmund, W. Germany, in June 1990. It presents the approach to VLSI design of Neural Networks developed by leading experts from Europe over the last couple of years.

The intention was to provide a collection of revised and extended papers which an interested VLSI-designer or student of Electrical Engineeringor Computer Science can use as an introduction to research and as a reference for further work by his own.

In Chapter 1 some important considerations preceding VLSI implementation of neural nets are recalled.

Chapter 2 gives a review of CCD technology and focusses particularly on the neural net potential provided by Junction-CCDs. Chapter 3 presents a full account of the possibilities for analog storage of adjustable synapses and discloses results recently obtained. The computational precision of analog design is dealt with in Chapter 4; a specific method is presented to reduce the errors due to mismatching of components in a VLSI neuron.

Chapters 5 and 6 cope with VLSI architectures which follow the biological models. Digital, analog and pulse stream techniques are discussed.

Chapters 7 to 10 are devoted to the presentation of various types of application specific neural chips. Pulse density modulated neural nets, adaptive associative memories and arbitrarily structured neural nets are supported by special silicon. Digital as well as analog design by means of a neural silicon compiler, gate arrays or full custom design are described.

Chapters 11 to 14 deal with the design of neurocomputers. Chapter 11 presents the fundamental ring systolic architecture and discusses system architectures using off-the-shelf DSPs. Architectural and implementation issues concerning the processor granularity of ring systolic systems are presented in Chapter 12. Chapter 13 discloses a detailed study of neural algorithms which results in a unified description of neural models by 3 equations. A related fine-grain neurocomputer architecture based on a VLSI Signal Processor is described in Chapter 14.

The book concludes with chapter 15 which contains the computer scientist's view on neurocomputer design. Software design and its implication on hardware are particularly emphasized.

Contents

GUIDE LINES TO VLSI DESIGN OF NEURAL NETS

U. RAMACHER

INTRODUCTION

The response and the characteristics of present models of artificial neural nets are primarily investigated by simulation on vector computers, workstations, special coprocessors or transputer arrays. The fundamental drawback of such simulators is that the spatio-temporal parallelism in the processing of information that is inherent to the neural net is lost entirely or partly and that the computing time of the simulated net especially for large associations of neurons (tailored to application-relevant tasks) grows to such orders of magnitude that a speedy acquisition of "neural" know-how is hindered or made impossible.

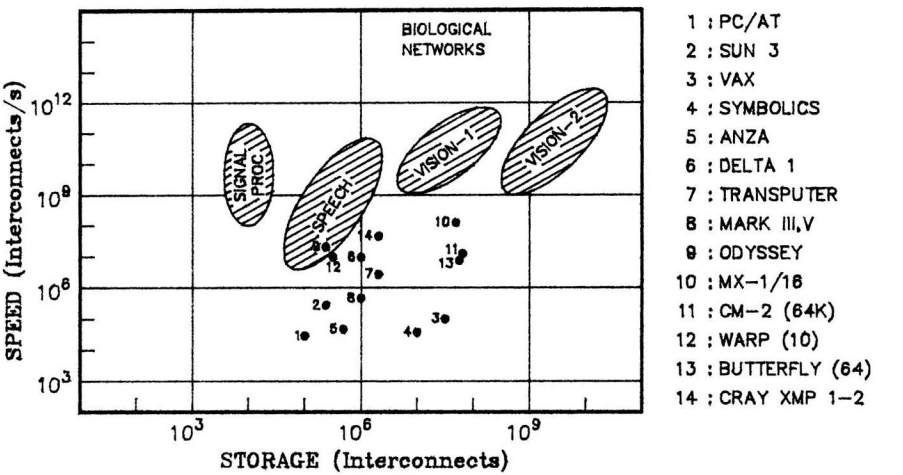

Figure 1 Computational capabilities of neural network simulators versus computational requirements of some applications

Figure 1 shows the performance obtainable with commercially available simulators [1] in terms of weights and weights per second. This must be confronted with the applicational needs. It becomes obvious that today's hardware capabilities are limiting the development of neural network research.

An appreciable reduction in computing time and thus the handling of largish tasks or those that are to be executed in realtime become possible with specially designed neural hardware. Apart from the shortest possible computing time, neural hardware offers a very much smaller

structural volume than can be implemented with simulators for the same task. This aspect is especially important when neural hardware is to be incorporated in terminals for man-machine communication or mobile robotics.

VLSI OPPORTUNITIES FOR NEURAL NETS

Depending on the application under consideration, the user will tell whether his problem is accessible by simulation on conventional computers or not.

If yes, the applicational task under consideration can be well defined and the user will specify the kind of data format and the degree of weight resolution, the size and type of the network and the processing speed for the recall mode. If the real-time requirement cannot be satisfied by software implementation, it makes sense to think about designing special hardware. Because the weights are computable in advance, there is no extra circuitry required other than for programmable or hard-wired weights and discrimination. The task of the architecture developer consists in this case of putting as many synapses as possible on the chip for a particular application, for the pattern storage capacity increases with the number of implemented neurons and the computing time reduces linearly with the number of implemented synapses. Considering just one applicational area, namely signal processing, it has been demonstrated that the number of synapses required is of an order that can be implemented on a single chip with today's technology [2,3] . For small-scale (in terms of the number of synapses required) applications there is thus the possibility of application-specific neural chips with programmable or fixed weights (see figure 2).

The learning algorithm (of an application which is accessible to simulation) only has to be considered in hardware terms if there is a relevant real-time requirement. The latter is imaginable, for instance, if the learnt knowledge is valid for a short time and new learning is repeatedly necessary. Obviously, supporting the learning will be at the expense of the number of synapses implemented, and it has to be checked whether single-chip integration will be possible at all. Wafer Scale Integration may turn out to provide the integration potential for computing and storing synaptic weights and intermediate results produced by the learning algorithm [4,5]. An alternative popular proposal is to distribute the implementation of a neural net plus learning algorithm over several chips and cascade these.

Neural nets for applications like vision or speech, on the other hand, overtax the single-chip integration potential of present technology as well as that of future 0.3-μm technology by whole orders of magnitude

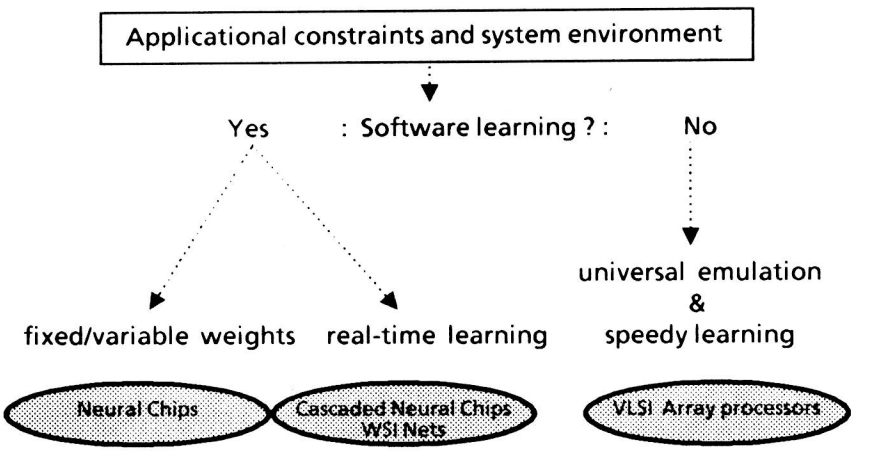

Figure 2 Applicational dimensions and VLSI opportunities

(see figure 1). Particularly the weights will have to be stored off-chip. The size of such a net will not permit simulation (especially of the learning phase) within a reasonable period of time and therefore the weights cannot be determined by simulation. This means that little or no engineering know-how has been accumulated for these large applications. Consequently, the VLSI architecture must be designed for universal emulation of neural network structures and speedy learning. VLSI architectures of this type will obviously look different from those considered in small-scale applications.

In the sequel the two opportunities and tries to present a guide line to neural VLSI design with special focus on the requirements caused by applicational needs, technological constraints and system environment.

SINGLE-CHIP INTEGRATION

Single-chip integration of a neural net means on-chip storage of the weights (not more than a few ten thousands of synapses today [2]). Consequently the amount of information a neural chip can process will be limited in size. This means in turn that the application must match the implementation potential of the technology. Fortunately, the learning for neural nets that can still be integrated on a chip can be performed on conventional computers with reasonable time expenditure, so the applicational task under consideration can be well defined and the hardware matched optimally to the task.

On the one hand, on-chip storage of weights offers an easy way to achieve real-time action by neural networks, since there exists no pad-

bandwidth problem for the weights. On the other hand, the VLSI designer faces the interconnection problem, irrespective of which design style (analog, binary, digital) or which technology is used [6,7]. If several ten thousands of weights were to be connected physically to a neuron and some thousand neurons to be implemented, the wiring area would grow to such orders that the delay on the wires tends to exceed the latency time of the functional block representing a neuron.

In principle, there are two ways to fight the interconnection problem: firstly, by reducing the technological structure size, and secondly, by architectural means. The first way dominated the early era of neural hardware design [8,9,10] . Nowadays, the second course is followed preferably, since an architectural solution of the interconnection problem is the cheapest. As a rule, a designer should check first for the required processing time (realtime conditions) of the application under consideration and then think out to what extent it is possible to deviate from the ideal massive parallel networking of a neural net. Together with the decision as to whether analog or digital signal processing concepts are to be applied and the selection of a technology, there results an initial architecture draft. Instead of reviewing architectures the following sections show what influence the technology and signal processing concept exert on the further design of an architecture whatever it may look like.

Analog Design Issues

Particularly for analog realization, the emphasis of circuit design is on the exploitation of the functional properties inherent in basic circuit elements. Generally the architecture draft needed for the analog implementation of a neural net will concentrate on representing the synaptic weighting, the neural ignition response (discriminator function) and the controlling of data input and output. An important requirement is the compactness of the connection element, because the cell size mainly determines the overall area of the network.

An advantage of analog circuits in comparison to digital circuits is that they can "process more than 1 bit per transistor". If this benefit is to be made use of, however, the following problems have to be mastered:

- Nonvolatile storage of analog weights provides very high synaptic density, but may be not sufficiently often programmable [11].
- The design of a synapse, the size of a neural network and the degree of analog resolution are dependent on each other [12].
- A major design problem with analog circuits is the relation of accuracy to chip area. The more precisely one wishes to control the matching of the analog components, the more chip area is needed. An analog depth of not more than 8 bit is recommendable. Crosstalk

and susceptibility to coupled-in interference make special precautions necessary for analog signal processing.

-- The minimal chip area is also influenced by factors like noise and current consumption. Low current consumption calls for high-valued resistors in resistor-capacitor circuitry. Low noise creates certain limits for minimal transistor Q surfaces and capacitors; this applies to switch-capacitor as well as to resistor-capacitor technique.

-- The temperature dependence, clock feed-through and process-parameter dependence can be reduced in analog circuits so that they no longer interfere, but this is done at the expense of circuit complexity and has an effect on the chip area.

-- With future 0.3-µm transistor channel length, a lower supply voltage than 5 V must be expected, so questions of low-voltage design have to be considered. Accurate transistor modelling, innovative circuit techniques and design cleverness will be significant here like in the cases above.

The limited precision in grading the weights (realised, for example, in the form of ohmic resistances or switched capacitors) means, on the one hand, limited computing accuracy for an analog implementation and therefore influences the number and complexity of the patterns that can be reliably processed with an analog net. This applies equally to the selection of the discriminator: the computing accuracy of the entire analog chip has to be considered.

The limited precision in processing information by the analog neural net means, on the other hand, that the information must be encoded in a redundant or fuzzy fashion, i.e. only as sharply defined as is necessary for secure recognition.

If a learning algorithm (with its multiple iterations) is to be implemented in analog circuitry, it is necessary to ensure a fortiori whether the intended application can at all be learnt.

Therefore it is the task of the user concerned with problem analysis and modelling to characterize those tasks of pattern processing in which deviations of the actual weight values from the pre-computed ones can be tolerated and where it is possible to make do, for example, with ternary weighting of binary input/output signals. For, the latter poses the least problems to the analog designer.

In the analog implementation of neural nets it is consequently a matter of bringing together the application-oriented problem analysis and the circuit architecture; this is the only way to determine the application spectrum that can be implemented with analog hardware. Isolated definition of the effort for analog multiplication, for instance, would be just as inappropriate on the part of the chip architect as determination of

the weights by simulation, without allowing for the computing accuracy that can be achieved, on the part of the user.

Digital Design Issues

What is of clear advantage here is that the application-oriented problem analysis and modelling of net characteristics can be made independently of circuit design. For by simulation one can determine the computation accuracy (word width) for each function block before implementing it. The technology and the circuit architecture only determine how many neurons can be integrated on a chip and what processing time can be achieved.

It is the task of the chip designer then to implement the neural algorithms in accordance to the state-of-the-art in Digital Signal Processing. A point to be tackled, for instance, is the problem of the so-called limit cycles, i.e. parasitic oscillations. These can appear in recursive digital structures (in the learning phase or in feedback nets for example) as a result of amplitude quantization (in analog recursive structures this problem consequently does not occur). With the neurons the word limiting, for algorithmic reasons, has to be implemented as a saturation characteristic anyway, so there is no danger of large limit cycles. The small limit cycles, which can be produced by bit manipulations (rounding, truncation), are dependent on the word lengths used and the structure of the feedback net. The problem of limit cycles can be eliminated however by appropriate effort in computing accuracy and numeric range (floating point). If this effort becomes unsupportable, it will be necessary to check whether such limit cycles appear and whether they can have a disturbing effect. One should also investigate whether limit cycles can be suppressed by selecting appropriate algorithms and measures.

The finer resolution of the weights, which is easy to implement in digital architecture, allows more individual weighting of the input signals than in analog processing. Furthermore, digital design can supply much easier the word width necessary to implement a learning algorithm. Thus it can be concluded that digital neural chips take on a field of application in their own right.

Analog versus Digital

Figure 3 displays the various ways how analog, binary or digital data can be processed. As discussed above, if the information must be processed with high precision (say with not less than 8 bits) or learning is to be supported on-chip, digital circuitry is the right candidate for implementing a neural net; conversely, for applications which do not need hardware support for learning and for less severe requirements in computation precision, analog design seems to dominate.

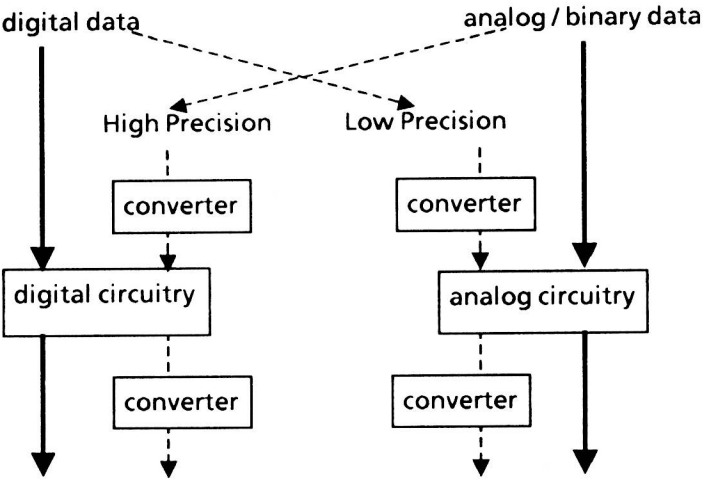

Figure 3

This picture changes if the costs for interfacing the data with the neural circuitry are to be taken into account. For instance, video images with, say, 8 bit pixels could be operated on by analog circuitry. But this would require fast D/A converters which were to be provided at the input and output ports. The costs of these could speak in favour of an all digital design [2].

Also, if the requirements for computing accuracy are not too high, various coding and transmission methods can be used that offer high noise immunity, e.g. pulse-density modulation or pulse-width modulation. But then various converters are indispensable. The effort for such complex circuit blocks and for pads can be kept within supportable limits through the use of multiplexers. However the use of time-division multiplex with continuous-time circuits (i.e. RC circuits) is not meaningful if the resistance of the switches affects the function. In this case, time-division multiplex is much easier to implement with discrete-time circuits , i.e. analog switch-capacitor circuits or digital circuits.

Meanwhile, a great number of technologies have been tried for direct implementation of neural nets, and a dozen chips built [13] . To get an idea of what processing speed and net size can be achieved by single-chip implementation some recent examples are reviewed.

An example for mostly digital design is presented by the configurable CMOS neural network chip described by H. P. Graf [2]. The chip provides 256 blocks each of which consists of 128 binary synapses. The ternary weights are stored in a 6-transistor cell and the multipliers are realized as inverted XORs. The multipliers´ outputs are accumulated in analog